CLAIMS

What is claimed is:

1. A multi-mode texture decompression method for use during graphics processing, comprising;

(a) sending a request for compressed texture data to memory;

(b) receiving the compressed texture data from the memory;

(c) identifying at least one of a plurality of compression algorithms associated with the compressed texture data; and

(d) decompressing the compressed texture data in accordance with the identified compression algorithm.

2. The method as recited in claim 1, and further comprising prior to sending the request, compressing the texture data utilizing the plurality of compression algorithms, selecting the most favorable compressed texture data, and storing the most favorable compressed texture data in the memory.

3. The method as recited in claim 2, wherein the most favorable compressed texture data is the most accurate replication of an original version of the texture data.

4. The method as recited in claim 2, and further comprising storing a mode identifier with the compressed texture data.

5. The method as recited in claim 4, wherein the at least one of the plurality of compression algorithms associated with the compressed texture data is identified utilizing the mode identifier.

6. The method as recited in claim 5, wherein the mode identifier includes a mode bit.

7. The method as recited in claim 1, wherein at least one of the compression algorithms represents a 4x4 block of texels of the texture data utilizing two bits per texel only if the textels are opaque, each 4x4 block of texels including two 16-bit colors stored in an RGB 565 format and two additional colors created by interpolating between the two 16-bit colors stored in the RGB 565 format to form a 4-entry lookup table, where a 2-bit index is adapted for being used to determine which 16-bit color from the lookup table is used for each texel of the 4x4 block of texels, and transparent texels are represented by making one of the four 16-bit colors transparent.

8. The method as recited in claim 1, wherein at least one of the compression algorithms represents a 4x8 block of texels utilizing three bits per texel, each 4x8 block of texels including two 15-bit colors stored in an RGB 555 format and five additional colors created by interpolating between the two 15-bit colors stored in the RGB 555 format to form an 8-entry lookup table, where an eighth 15-bit color is defined to be a transparent color, and a 3-bit index is used to determine which 15-bit color from the lookup table is used for each texel in the 4x8 block of texels.

9. The method as recited in claim 1, wherein at least one of the compression algorithms represents a 4x8 block of texels utilizing two bits per texel only if the textels are opaque, each 4x8 block of texels including four 15-bit colors in an RGB 555 format to form a 4-entry lookup table, a 2-bit index is adapted for being used to determine which of the four 15-bit colors is assigned to each texel.

10. The method as recited in claim 1, wherein at least one of the compression algorithms represents a 4x8 block of texels by two bits per texel, each 4x8 block of texels including three 20-bit colors stored in a 5555 format, where a first and second one of the 20-bit colors are used for primary colors of a left 4x4 sub-block of the 4x8 block of texels, and a second and third one of the colors are used for primary colors of the right 4x4 sub-block of the 4x8 block of texels, where two additional 20-bit colors are created in each 4x4 sub-block of texels by interpolating between the 20-bit colors associated with the corresponding 4x4 sub-block of texels, where a 2-bit index is adapted for being used to determine which of the four 20-bit colors is assigned to each texel, and a lookup table is used to determine which 20-bit color is applied to each texel.

11. A multi-mode texture decompression computer program product for use during graphics processing, comprising;

(a) computer code for sending a request for compressed texture data to memory;

(b) computer code for receiving the compressed texture data from the memory;

(c) computer code for identifying at least one of a plurality of compression algorithms associated with the compressed texture data; and

(d) computer code for decompressing the compressed texture data in accordance with the identified compression algorithm.

12. A multi-mode texture decompression system for use during graphics processing, comprising;

(a) a texture fetch module adapted for sending a request for compressed texture data to memory, and receiving the compressed texture data from the memory;

(b) a format detection module adapted for identifying at least one of a plurality of compression algorithms associated with the compressed texture data; and

(c) a plurality of decompression modules coupled between the texture fetch module and the format detection module, the decompression modules adapted for decompressing the compressed texture data in accordance with the compression algorithm identified by the format detection module.

13. The system as recited in claim 12, wherein at least one of the compression algorithms represents a 4x4 block of texels of the texture data utilizing two bits per texel only if the textels are opaque, each 4x4 block of texels including two 16-bit colors stored in an RGB 565 format and two additional colors created by interpolating between the two 16-bit colors stored in the RGB 565 format to form a 4-entry lookup table, where a 2-bit index is adapted for being used to determine which 16-bit color from the lookup table is used for each texel of the 4x4 block of texels, and transparent texels are represented by making one of the four 16-bit colors transparent.

14. The system as recited in claim 12, wherein at least one of the compression algorithms represents a 4x8 block of texels utilizing three bits per texel, each 4x8 block of texels including two 15-bit colors stored in an RGB 555 format and five additional colors created by interpolating between the two 15-bit colors stored in the RGB 555 format to form an 8-entry lookup table, where an eighth 15-bit color is defined to be a transparent color, and a 3-bit index is used to determine which 15-bit color from the lookup table is used for each texel in the 4x8 block of texels.

15. The system as recited in claim 12, wherein at least one of the compression algorithms represents a 4x8 block of texels utilizing two bits per texel only if the textels are opaque, each 4x8 block of texels including four 15-bit colors in an RGB 555 format to form a 4-entry lookup table, a 2-bit index is

adapted for being used to determine which of the four 15-bit colors is assigned to each texel.

16. The system as recited in claim 12, wherein at least one of the compression algorithms represents a 4x8 block of texels by two bits per texel, each 4x8 block of texels including three 20-bit colors stored in a 5555 format, where a first and second one of the 20-bit colors are used for primary colors of a left 4x4 sub-block of the 4x8 block of texels, and a second and third one of the colors are used for primary colors of the right 4x4 sub-block of the 4x8 block of texels, where two additional 20-bit colors are created in each 4x4 sub-block of texels by interpolating between the 20-bit colors associated with the corresponding 4x4 sub-block of texels, where a 2-bit index is adapted for being used to determine which of the four 20-bit colors is assigned to each texel, and a lookup table is used to determine which 20-bit color is applied to each texel.

17. A multi-mode texture decompression system for use during graphics processing, comprising;

(a) means for sending a request for compressed texture data to memory;

(b) means for receiving the compressed texture data from the memory;

(c) means for identifying at least one of a plurality of compression algorithms associated with the compressed texture data; and

(d) means for decompressing the compressed texture data in accordance with the identified compression algorithm.

18. A multi-mode texture compression method for use during graphics processing, comprising;

(a) compressing texture data utilizing a plurality of compression algorithms;

(b) selecting the most favorable compressed texture data;

(c) storing the most favorable compressed texture data in memory;

(d) storing a mode bit with the most favorable compressed texture data in the memory;

(e) sending a request for the compressed texture data to the memory;

(f) receiving the compressed texture data from the memory;

(g) determining the mode bit associated with the received compressed texture data;

(h) identifying at least one of the plurality of compression algorithms associated with the compressed texture data based on the mode bit; and

(i) decompressing the compressed texture data in accordance with the identified compression algorithm.

19. A multi-mode texture compression method for use during graphics processing, comprising;

(a) compressing texture data utilizing a plurality of compression algorithms;

(b) selecting the most favorable compressed texture data;

(c) storing the most favorable compressed texture data in memory; and

(d) storing a mode bit with the most favorable compressed texture data in the memory;

(e) wherein the mode bit associated with the received compressed texture data is capable of being used to identify at least one of the plurality of compression algorithms associated with the compressed texture data such that the compressed texture data is capable of being decompressed in accordance with the identified compression algorithm.

20. A data structure stored in memory for compressing texture data representing a YxZ block of texels utilizing three bits per texel, each YxZ block of texels including two X-bit colors stored in a predetermined format and five additional colors created by interpolating between the two X-bit colors stored in the Predetermined format to form a lookup table, where an eighth X-bit color is defined to be a transparent color, and a W-bit index is

used to determine which X-bit color from the lookup table is used for each texel in the YxZ block of texels.

21. A data structure stored in memory for compressing texture data representing a YxZ block of texels utilizing two bits per texel only if the textels are opaque, each YxZ block of texels including four X-bit colors in a predetermined format to form a lookup table, a W-bit index is adapted for being used to determine which of the four X-bit colors is assigned to each texel.

22. A data structure stored in memory for compressing texture data representing a YxZ block of texels by two bits per texel, each YxZ block of texels including three X-bit colors stored in a predetermined format, where a first and second one of the X-bit colors are used for primary colors of a left YxY sub-block of the YxZ block of texels, and a second and third one of the colors are used for primary colors of the right YxY sub-block of the YxZ block of texels, where two additional X-bit colors are created in each YxY sub-block of texels by interpolating between the X-bit colors associated with the corresponding YxY sub-block of texels, where a W-bit index is adapted for being used to determine which of the four X-bit colors is assigned to each texel, and a lookup table is used to determine which X-bit color is applied to each texel.